Spectrum Profi Club

für alle Spectrum und SAM Freunde



Wann wird's mal wieder richtig Sommer...?
(vielleicht kommt er ja auch so spät wie dieses Info?)

Wolfgang & Monika Haller, Tel. 0221/685946
Im Tannenforst 10, 51069 Köln
Bankverbindung: Dellbrücker Volksbank
BLZ 370 604 26, Konto-Nr. 7404 172 012

Ausgabe 91

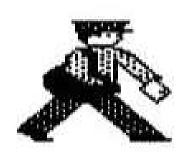
Juli 1997

KALEIDOSKOP

... unter dieser Überschrift wollen wir künftig ein buntes Allerlei aus der Spectrum- und SAM-Welt zusammenstellen. Eure Informationen und Kommentare sind dabei herzlich willkommen. Fangen wir also an mit:

DER SPC IST NUN VÖLLIG UMGEZOGEN!
Diese Nachricht sollte eigentlich für euch treue Leser nichts neues sein. Für unseren Postboten indessen ist sie es wohl. Deshalb unsere Bitte: Schreibt immer unsere Namen mit auf den Umschlag, und nicht nur SPC oder WoMo-Team. Hier nochmal unsere vollständige neue Anschrift:

Spectrum/SAM Profi Club
Wolfgang und Monika Haller
Im Tannenforst 10
D-51069 Köln



PS: Unsere Telefonnummer haben wir behalten, ein Vorteil, wenn man innerhalb eines Ortes umzieht.
Ein Dankeschön an dieser Stelle an alle, die uns bei unserem Umzug geholfen haben. Und auch an diejenigen, die uns in ihren Briefen "moralisch" unterstützten. Es kann ab jetzt nur besser werden.

DAS TREFFEN IN MÖNCHENGLADBACH RÜCKT NÄHER...
und uns liegen schon einige interessante Zusagen vor. Vor allem aus Kreisen des ZX-Teams, wo für unser Treffen kräftig die Werbetrommel gerührt wurde. Allerdings sickerte aus einer verlässlichen Quelle (Leider dürfen wir den Namen des Informanten nicht preisgeben, nur sein Synonym: China-Man.) durch, das man, ähnlich wie damals bei Thyssen und Krupp, eine klammheimliche Übernahme (Infiltration!) des SPC plane. Nun appellieren wir natürlich an euch: Kommt in Scharen. Je mehr wir sind, desto besser können wie einen solchen Versuch abblocken (wie beim Spiel "Risiko").
Auch Ronald Raaijen aus den Niederlanden hat sich gemeldet und läßt über uns mitteilen, das er einen Sinclair PC 200 und einen ZX81 mit Diskinterface und Mnemotech-Keyboard mitbringen wird. Desweiteren wird es ein Sortiment an Hardware Angeboten und Zeitschriften geben. Und für die Freunde des Spectrum-Emulators gibt es als besonderes Schmanckerl Software für den Emulator und vor allem Ronalds einzigartiges Spectrum D-Base.

Damit niemand sagen kann, er hätte (mal wieder) von nichts gewußt, hier nochmal Zeit und Ort des diesjährigen Treffens:



Samstag/Sonntag, 16. + 17. August 1997
Am Baumlehrpfad 2 (DRK-Haus)
41179 Mönchengladbach (Rheindahlen)

Also: Dick im Kalender markieren! Und vielleicht schon mal folgendes im Hinterkopf behalten:

ENLIGHT DEMO PARTY 1997
Joho, Post von Eugenijus Dalnikovas, der uns folgendes mitteilt:
"How are you? Sorry, but me and Sigis will not go to your party. It is a money problem. It's a pity :-(
I collected some money for the **Enlight 97** demo party. But there is no idea where to get money for your party.
Greetz to Monika. Have a nice party. After end of your party, Enlight will start! We'll present demos, music and gfx there!
Would you go to Enlight? And see the Russian demo scene in face?
Please answer faster and I'll send you infos! If you can't go, maybe some of your friends are able to come. C ya!?"

Soweit Eugenijus. Um eine schnelle Antwort bemüht, machen wir das gleich hier:

"Hello Eugenijus. Thank you for your letter and for the goodwill thinking to come to the SPC party. We understand fully, that it is a money problem, and believe us, our problems are equal. Two movings within a year can bring you to ruin. Don't think, that Germany is "the country of milk and honey". We know, that a lot of people here live at the edge of their existence minimum. But also for those with a bit more money isn't it easy to make a short step to visit Russia. However, we have printed your proposal and wish you any success. And who knows, life can be full of surprise."

Nun noch weitere Nachrichten. Zuerst eine gute, und die stammt ebenfalls von Eugenijus:

PENTAGON 128K ZU KAUFEN
Hello european dudes. Now you have a chance to get a Pentagon 128K + TR-DOS + AY 3-8910 for

price made specially for SPC members! Send 149$ USD to me and you'll get it as fast as possible! Send money inside a defekt disk. It is because of problems with the post in Russia. All pentagons are self-made and must be connected to TV or monitor by RGB, sync and ground.
And you'll get 5 disks with newest software from Russia for free.
DON'T MISS YOUR CHANCE!
In case of interest write to:

Dalnkikovas Eugenijus
Kalvariju g.142-3
2042 Vilnius
Lithauen/Lithuania

Sicher ein gutes, auf jeden Fall aber ein interessantes Angebot. Lediglich mit der Bezahlung hätten wir da so unsere Probleme. Geld in einer defekten Diskette zu verschicken erscheint uns nicht sehr verläßlich. Und was ist, wenn es wirklich verlorengeht? Es muß doch eine andere Lösung geben. Wir werden Eugenijus diesbezüglich fragen.

Im Mai-Info hatten wir die Antwort von Datel auf die Anfrage von Jean Austermühle veröffentlicht. Nun, vielleicht tut sich jetzt doch was:

HILFE BEIM PLUS-D NACHBAU?

Es sieht so aus, als ob wir unerwartet Hilfe im Bemühen bekommen, das Plus D nachzubauen. Denn wir erhielten einen Anruf aus England, in dem uns eine Größe aus der Szene Unterstützung zusagte. Nun wollen wir keine wilden Gerüchte in die Welt setzen, bevor uns nicht auch etwas in schriftlicher Form vorliegt, und das ist zu diesem Zeitpunkt noch nicht der Fall. Auf jeden Fall haben wir Jean (und Peter) schon über den Stand der Dinge informiert und denken, das wir zum nächsten Kaleidoskop schon wesentlich mehr wissen.

SPECTRUM-REPARATUREN

Leute paßt auf eure Geräte auf! Denn zur Zeit sind unsere Spezialisten Peter Rennefeld und Jean Austermühle beruflich so ausgelastet, das ihnen kaum Zeit für Reparaturen bleibt. Auch bei Fred Dürkes vom SUC ist laut Suc-Session nicht gewährleistet, das er Reparaturen ausführt. Es könnte also ein echter Engpaß eintreten, d.h. längere Wartezeiten. Wenn bei euch also ein echter "Notfall" eintritt, dann nehmt am Besten persönlich Kontakt mit den beiden auf, z.B. wegen eines Leihgerätes. Hier nochmal die Adressen:

Jean Austermühle, Sternwartstraße 69
40223 Düsseldorf, Tel. 0211/395460

Peter Rennefeld, Genhodder 19
41179 Mönchengladbach, Tel. 02161/571141

NEUE KATALOGE VON SINTECH

SINTECH hat neue Kataloge für den SPECTRUM, CPC, C 64, Atari ST und Atari 800/130 herausgebracht. Diese können kostenlos unter der Tel.-Nr. 0711/77503 angefordert werden.

SPC VOR DEM AUS???

In AlchNews war es zu lesen - und der SUC hat es aufgegriffen - das wir den SPC im April 1998 schließen werden. Thomas vermutet dahinter ein Mißverständnis. Was ist also an dieser Nachricht dran?
Einfach gesagt hat Andy Davis hier einen Gedankengang, den wir in einem Brief an ihn geschrieben hatten, als Tatsache veröffentlicht. Wir hatten ihm nämlich mitgeteilt, das die 100. Ausgabe (daher also April 1998) unseres Infos ein schöner Anlaß wäre, nach über 8 Jahren und Monat für Monat erscheinendem Info, einen Abschluß zu machen. Solche Gedankengänge hatten wir in der Vergangenheit auch schon mal.
Eines vorweg - es ist keine beschlossene Sache. Irgendwie weiß ich z.B. garnicht, wie man ohne den SPC leben kann? Und selbst wenn, dann würden wir allenfalls aufhören, wenn wir den Club danach in guten Händen weitergeleitet wüßten, schließlich sind wir nicht Spectrum UK. Also - Entwarnung!

LARRY LAFFER TEXTADVENTURE

Nele Abels-Ludwig, Fleißarbeiter beim SPC, hat ein neues Projekt beendet: Die Umsetzung des Nur-Text-Adventures "Leisure Suit Larry I" unter dem Namen "Larry Laffer Textadventure" für den Spectrum unter Verwendung des PAW (Quill-Nachfolger). Dieses Adventure läuft nur im 128K Modus - und zur Zeit auch nur auf dem Spectrum-Emulator. Um das Projekt zum Abschluß zu bringen, braucht Nele aber noch Hilfe aus dem Userkreis. Wir zitieren:
"Mein 'Larry Laffer Textadventure' ist fertig geschrieben und hat die Alpha-Testphase verlassen. Das bedeutet, ich bin nicht mehr in der Lage es schleifen zu lassen und als unvollendetes Projekt in der Ecke verstauben zu lassen. :)
Jetzt brauche ich möglichst viele Beta-Tester, und ich wäre euch sehr verbunden, wenn ihr einen Aufruf im Clubheft bringen könntet. Natürlich hat sich letztendlich doch noch ein Problem ergeben, ich schaffe es nicht, den Snap auf den echten Spectrum zu laden. Deshalb: Kennt ihr jemanden, der in der Lage ist aus dem Emulator heraus auf Tape zu speichern, am besten über das parallele Tape-Interface?"
Na, das sollte doch wohl möglich sein!!? Ich denke hier an unsere Emulator-Profis Bernhard Lutz, Claus Jahn, Matthias Wiedey (um nur einige zu nennen) und für die Beta-Testphase werden sich sicher auch Tester finden lassen (z.B. Double H Software?).
Wenn alles klappt, wird Nele auch zum Treffen erscheinen und dort sein Programm vorstellen.

Mitglied Nr. 109

Wir freuen uns immer wieder, wenn wir Briefe erhalten, in dem sich ein "verloren geglaubtes Schaf" zurückmeldet. So, wie in diesem Fall. Wir begrüßen also "welcome back" im Club:

Ingolf Fitzner, Talstraße 7
07407 Rudolstadt

Wir hoffen, das Du inzwischen alle Dir fehlenden Hefte erhalten hast und somit das Sommerloch stopfen kannst.

Ingolf hat auch gleich eine Frage an unsere **Beta-Disk-User:** Funktioniert das Beta-Disk-Interface auch am Spectrum+ 128K? Ich habe nämlich eins bekommen, aber es erfolgt keine Reaktion.

Also, welcher Beta-Spezialist kann Ingolf helfen?

Diesmal kommt der SAM, ebenfalls aus Zeitgründen wegen unseres Umzugs, etwas zu kurz. Für die Spielnaturen unter uns hat Nico Kaiser aus Ilmenau auch diesmal wieder einige Cheats auf Lager:

SAM: Cheats für "Swallower"

Wer mal wieder so richtig "Pacman" spielen will, dem sei das Spiel "Swallower" ans Herz gelegt. Swallower stammt von Relion Software und wurde auf "Enceladus 12" veröffentlicht. Es ist nicht einfach zu spielen (Steuerung), aber saugut. Insgesamt hat man 30 Level zu bewältigen, was kaum jemandem auf Anhieb gelingen dürfte. Aber es gibt einen (verborgenen) Cheatmodus, und zwar, wenn man folgende Eingaben in der Highscore-Liste macht:



Ganeymede (macht unsterblich)
Ghormenghast (ein schneller Geist)
Prunesquallor (zwei Geister)
Bacon (zwei schnelle Geister)

Also - am besten gleich ran ans Spiel und Ausprobieren. Viel Spaß dabei wünscht euch

Nico Kaiser, Geschwister-Scholl-Str. 11a
98693 Ilmenau, Tel. 03677/883785

Der SAM im Internet



Gute Nachricht: Auch der SAM hat inzwischen seinen Einzug ins Internet gefunden. Z.B. bei Steven Pick's SAM Coupe Bit oder ftp.nvg.unit.no.

Aber auch die, die noch nie einen SAM gesehen haben, können nun in den Genuß kommen, den SAM und einige Programme dazu kennenzulernen. Und zwar - wie sollte es anders sein - als Emulator auf den PCs.

Anders als beim Spectrum jedoch gibt es den SAM-Emulator nur in einer Unix-Version. Unix kann man sich ebenfalls downladen, aber was dann?

Deshalb unsere Frage: Wer von euch hat schon einmal den SAM-Emulator unter Unix (oder Linux) ans Laufen bekommen? Hat irgendjemand einen Tip dazu. Oder gibt es gar inzwischen eine Windows oder Windows 95 Version?

Oder haben wir gar jetzt erst den einen oder anderen neugierig gemacht? Wir freuen uns über eure Erfahrungen oder Berichte. Das WoMo-Team

---

Hilfe, mein 128er spinnt!!

Rupert Hoffmann hat ein Problem mit seinem 128er. Obwohl er eine neue Tastaturfolie eingelegt und auch den Bus gereinigt hat, fallen bei ihm immer wieder bestimmte Tasten bzw. Tastenkombinationen aus. Dies geschieht hauptsächlich im "Wordmaster", z.B. die Gänsefüßchen, Extended- und auch Graph-Mode. Beim normalen Programieren passiert dies jedoch nicht. Wer Rupert Hoffmann sagen kann, woran das liegen könnte, der kann mit ihm Kontakt aufnehmen unter seiner richtigen (!) Adresse (nicht die aus der Mitgliederliste 97):

Rupert Hoffmann, Tulpenstraße 22
92637 Weiden, Tel. 0961/6341321

PS.: Keine Sorge Rupert, der DTP-Kurs wird von mir fortgesetzt, sobald ich wieder etwas Luft habe. Wenngleich die Reaktionen sehr spärlich waren, weiß ich doch, das Informationen und Tips zu diesem tollen Programm immer gerne "aufgesogen" werden. Also bitte noch ein wenig Geduld. Wo von WoMo

Schickt euch doch mal e-mails...

Helge Keller hat sich mal wieder gemeldet. Unser Opus-Spezialist ist inzwischen auch "Opfer" seines Berufes geworden, der ihm kaum noch Zeit für sein Hobby läßt, was er aber ändern möchte. Helge fand übrigens den Aprilartikel von Heinz Schober genial :-) und teilte uns gleichzeitig seine e-mail Adresse mit:

keller@mahpl.mathematik.uni-karlsruhe.de

Dies brachte uns auf die Idee, mal eure Postkarten durchzugehen und nach weiteren, bisher nicht veröffentlichten e-mail Adressen von Mitgliedern zu schauen. Zweimal wurden wir fündig:

Wolfgang Krille: wkrille@t-online.de
Emil Obermayr: nobs@welfen-netz.com

Wer seine e-mail Adresse ebenfalls hier veröffentlicht sehen möchte, nichts einfacher als das: schickt uns eine Postkarte. Vielleicht entsteht auf diese Art ja eine neue Art der Kommunikation unter den Speccy-Freunden.

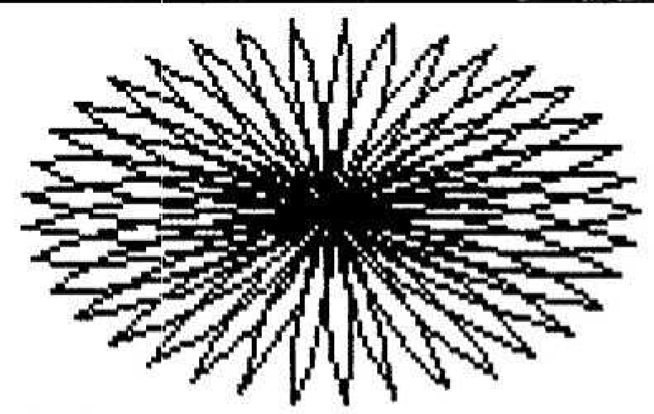
In der Kürze liegt die Würze

oder 1 KByte ist genug (Teil 4)

Ja, das letzte Mal haben wir "Spielereien" mit dem Border gemacht. Wer das Programm "Effekt-show" ausprobiert hat, falls es überhaupt jemand getan haben sollte, sah das der Border mitsamt dem Paper in Action versetzt wurde; mit Farben, die der Spectrum normalerweise nicht aufweist.
Dies ist natürlich ein guter Übergang, so daß wir jetzt auf dem Paperbereich kommen. Gut, es gibt gute Grafikprogramme dafür; ich selbst benutze das Advanced OCP Art Studio 128. Am Joystickport der OPUS ein Mausadapter, und daran die Geos-Maus.
Aber jetzt wollen wir über einfache Programmierung reden, sonst ist es ja eine Themaverfehlung.
Normalerweise bietet der Bildschirm 22 Zeilen mit je 32 Zeichen an. Die Bildpunkte können auch einzeln angesprochen werden.
Dies ergibt dann 176*256 Punkte. Und hier kommt dann das einzigartge tolle Sinclair-Basic zum Einsatz. Dies sind folgende Wörter, wie z.B. **PLOT, DRAW, POINT, CIRCLE...**
Durch **CIRCLE 127,87,35** kann man einen schönen Kreis zeichnen; 127,87 als Mittelpunktkoordinaten und 35 ist der Radius. Vor etlichen Jahren machte ich eine Bekanntschaft, der den Commodore 64 hatte. Um einen Kreis zu zeichnen benötigte er sage und schreibe 50 Zeilen. Das ist ja grausam. Auch die ehemalige Zeitschrift Happy Computer testete einige Computer in Sachen Grafik. Beim Commodore Grafikfähigkeit gut und unter dem Aspekt Grafikprogrammierung: katastrophal.
Aber lieber zurück zu unserem geschätzten Specci. Durch die einfache Programmierung macht es so richtig Spaß zu experimentieren. Gebt doch einfach die folgende Zeile ein:

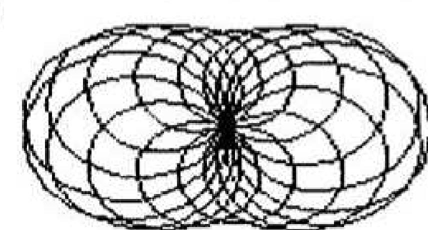
```
PLOT 128,0: DRAW 0,175,X*PI
```

Für x irgendeine **ungerade** Zahl eingeben. Mit Plot wird ein Punkt an eine bestimmte Adresse des Bildschirms erzeugt. 128 ist horizontal; die 0 ist vertikal. Als ungerade empfehle ich euch mal 255.
Eine "verbesserte" Variante erseht ihr aus dem Listing 1. Es hat lediglich eine Länge von sage und schreibe 411 Bytes. Ihr seht daraus, daß der Specci ein wahres Grafikwunder ist. Das Listing 2 ist noch ein ganz besonderes Schmankerl. Es ist allerdings nicht von mir, es stammt aus dem Buch "Sinclair ZX Spectrum Basic Handbuch" von D. Herget. Dieses Listing (458 Bytes) zeigt euch auch deutlich, wie gut POINT einsetzbar ist.



LISTING 1

```
 10 REM ***Rosettengrafik***
 20 INPUT AT 0,0;"Hoehe:",Hoehe,
    "Breite:",Breite
 30 INPUT AT 0,0;"Anzahl:",Anzahl,
   "Kruemmung:",Kruemmung
 40 FOR a=1 TO Anzahl
 50 LET b=a*2*PI/Anzahl
 60 LET c=1
 70 PLOT 127,87
 80 DRAW c*Breite*COS b,c*Hoehe*SIN
    b,Kruemmung
 90 DRAW -c*Breite*COS b,-c*Hoehe*SIN
    b,Kruemmung
100 NEXT a
110 STOP
120 GO TO 10
```



Nur 2 Beispiele von den vielen Möglichkeiten des Listings 1

LISTING 2

```
 10 REM ***POINT***
 20 FOR i=0 TO PI*.5 STEP .1
 30 PLOT 128,88: DRAW 127*SIN i,85*
    COS i
 40 PLOT 128,87: DRAW 127*SIN i,-85*
    COS i
 50 PLOT 127,87: DRAW -127*SIN i,-85*
    COS i
 60 PLOT 127,88: DRAW -127*SIN i,85*
    COS i
 70 NEXT i
100 FOR i=0 TO 255
110 LET k=0
120 FOR j=0 TO 175
130 IF POINT (i,j)=1 THEN  LET k=k=0:
    REM Schalter 0/1
140 IF k=1 THEN  PLOT i,j
150 NEXT j
160 NEXT i
```

So, das wär's für heute...

Rupert Hoffmann, Tulpenstr.22
92637 Weiden, Tel. 0961/6341321



Hallo Abenteurer!!

Als Computerinteressierte wißt Ihr natürlich was ein WARP ist. Nein, nein, keine Angst. Wir sind nicht in die PC-Welt abgedriftet noch, wollen wir uns hier stark machen für das beste Betriebs-system hinter dem des Spectrum. Wir werden heute einen Timewarp benutzen um in die Vergangenheit zu reisen, ganz genau in das Jahr 1983. Denn genau in diesem Jahr erschien das Programm "**Temple of Vran**" auf dem Soft-waremarkt der damals noch recht jungen Spectrum Fangemeinde. Dieses Adventure stammt aus dem Zyklus der Mountains of Ket Trilogie und bildet dort den zweiten Teil. Gemäß der damaligen Programmgestaltung ist es noch relativ langsam in der Verarbeitung der Eingaben, was manchmal etwas störend wirkt. Aber als wahrer Adventurer muß man ja beharrlich sein in seinen Bemühungen und deshalb dürfte es keine zu große Geduldsprobe werden. Aufgabe des Spielers ist es, diverse Edelsteine zu sammeln und an den richtigen Stellen abzulegen, damit wir Delphia letztendlich doch noch besiegen können. Aus-serdem wartet das Programm mit ein paar Kampfübungen auf, deren Ausgang zufallsbedingt ist, weshalb man nicht sagen kann wer gewinnt. Die Erfahrung zeigt aber, daß wir meist am längeren Hebel sitzen. Der Aufbau der Locations ist für ein Adventure dieses Umfangs recht klar und übersichtlich ausgefallen. Im einzelnen tref-fen wir folgende Locations und Gegenstände an:

01] In a passage
02] Outside the passage / sharp axe
03] On the bank of a wide river
04] On the edge of a large wilderness area / gnarled club
05] Amongst some low hills / bamboo cane
06] At the south of a group of hills
07] On a rough terrain / sleeping kitten, small trampoline
08] On sacred ground / wart
09] Near a door to the south
10] In a large room
11] In a wart kitchen
12] Near stairs going up
13] In a small connection passage
14] In a room with entrances north and west / skeleton, ruby
15] In a store room / peanuts
16] In an abandoned field
17] On a barren plain / elephant
18] In a field / parchment
19] On a small hill
20] On a path leading east / small mouse
21] At the top of a cliff / some rope
22] At the base of a cliff
23] Half way down the gorge
24] Near the end of the gorge / sapphire
25] At the lava river / stepladder
26] At a Dead end
27] In a small cubby hole / bow, coil of vine
28] In a large pit of quicksand
29] To the south of a grove of trees / emerald
30] In a circle of trees
31] In the west end of a small wood
32] In the east end of the wood / bit of horse hair
33] In an open field
34] In a low stone building
35] Near smashed metal barrels
36] In the middle of a road
37] At the west end of the road
38] In a high building A / monkey wrench
39] At the east end of the road / slab
40] In a high building B
41] At a bare landscape
42] In the entrance area to a huge dome
43] At a junction
44] At a T-junction
45] In the northern room / jet boots
46] At the bottom of the stairs
47] In a sleep room A
48] In the armoury / blaster
49] In a sleep room B / topaz
50] In a guard room
51] In a cloak room / mask
52] At the northern of a passage
53] In a small store room
54] In an even smaller store room / cassette tape
55] Amongst some low rocks / tape recorder
56] In a disorganised camp
57] At the east side of the lava / pair of gloves, ruby
58] In front of two huge stone doors
59] In the temple waiting room
60] Room with scorched walls

61) Room with paintings on the wall / monk
62) In a red room
63) In a yellow room
64) In a green room
65) In a blue room
66) In an altar room
67) In Delphia's bedchamber / sceptre
68) The End!!



Soweit die Locations zum beigefügten Plan. Kommen wir nun ohne Umschweife zur schrittweisen Lösung. Wir stehen an unserem Ausgangsort - In a passage......

E, get axe, E, chop tree, drop axe, E, get club, E, S, E, get kitten, W, S, kill wart (auf keinen Fall die Escape Option wahrnehmen!!), S, hit door, S, wake kitten, drop kitten, E, D, S, prepare club, S, hit skeleton (wiederum nicht die Escape Option wahrnehmen!!), get ruby, W, get peanuts, E, N, N, U, W, wake kitten, N, N, N, N, N, W, E, E, get parchment, read parchment, drop parchment, S, S, get trampoline, N, E, E, feed elephant, tie rope, D, S, S, drop ruby, drop trampoline, S, get stepladder, N, N, N, U, W, W, W, S, S, S, S, wake kitten, S, drop stepladder, N, wake kitten, E, D, S, S, W, open trapdoor, get stepladder, E, N, drop stepladder, climb stepladder, U, get bow, get vine, D, get stepladder, N, U, W, N, N, N, N, get cane, N, N, tie vine, fire bow, swing, drop bow, drop club, drop sword, drop ring, drop stepladder, N, N, press any key, E, get hair, wear hair, W, S, E, W, S, S, W, S, get wrench, N, E, E, get slab, S, drop hair, give wrench, E, U, E, N, get boots, wear boots, S, W, D, D, W, W, get blaster, E, E, E, get topaz, W, U, U, E, S, fire blaster, S, get mask, N, N, W, N, fire blaster, N, E, get tape, W, S, S, D, wear mask, W, W, get recorder, insert tape, play recorder, W, give recorder, press any key, drop boots, drop blaster, drop mask, get sword, get stepladder, get emerald, S, S, S, E, E, get mouse, E, D, S, S, drop stepladder, get ruby, throw ruby, get sapphire, climb stepladder, jump down, S, unlock door, drop key, S, E, cover hole, W, N, N, get ruby, get gloves, S, S, E, prepare sword. An dieser Stelle ist es ratsam den Spielstand abzuspeichern da es jetzt dann gleich zum Kampf mit einem Mönch kommt. Wir werden zwar wahrscheinlich gewinnen, aber sicher ist sicher. E, kill monk (die Sache mit der Escape Option ist ja schon bekannt), U, drop ruby, W, drop topaz, S, drop emerald, E, drop sapphire, W, S, S, drop mouse, wear gloves, get sceptre, N, N, N, E, D, W, W, insert sceptre, S, press any key....

Temple of Vran ist gelöst. Wir bekommen nun noch ein Passwort für den dritten Teil und das war es dann auch schon wieder.

Soviel dazu und bis bald.......

Harald R. Lack, Heidenauer Str. 5, 83064 Raubling
Hubert Kracher, Starenweg 14, 83064 Raubling

Time travel
swing
jump

Double 'H'
mapped by
Software

Fraktale Grafiken

für Basic-Anfänger (Teil 3)

In der letzten Folge haben wir herausgefunden, wie man einen einzelnen Punkt der Mandelbrotmenge iteriert, und wie man feststellt, ob sein Betrag im Verlauf mehrerer Iterationen gegen unendlich geht oder nicht. Daraus läßt sich die Farbe eines Punkts in der fertigen Grafik ermitteln. Aber ein einzelner Punkt ist noch keine Grafik... Wie erweitern wir also die Routine für einen einzelnen Punkt auf zwei Dimensionen? Die Antwort darauf ist eigentlich ganz einfach: Mit Hilfe von zwei FOR-NEXT Schleifen, also Zählerschleifen, wird Schritt für Schritt ein Rechteck abgetragen und ausgerechnet. Soweit so gut. Aber bevor wir das in den Spectrum einprogrammieren, müssen wir uns über zwei problematische Fragen klar werden. Erstens, wie stellen wir die Mandelbrotgrafik auf dem Bildschirm des Spectrum dar? Und zweitens, wie speichern wir die errechneten Werte im Spectrum?

Zunächst zum ersten Punkt. In den beiden letzten Folgen habe ich etwas vollmundig davon geredet, daß üblicherweise die errechneten Iterationen mit verschiedenen Farben dargestellt werden. Aber mit den Farben ist das auf dem Spectrum bekanntermaßen ja so eine Sache. Jeweils 8*8 Bildschirmpunkte teilen sich ein Byte im Attributspeicher, haben also eine gemeinsame Vorder- und Hintergrundfarbe. Wir könnten eine farbige Mandelbrotgrafik also nur in der Auflösung des Attributspeichers, d.h. in einer Auflösung von nur 32*22 Punkten darstellen, was natürlich nicht besonders befriedigend ist. Wir müssen also in den sauren Apfel beißen und Mittel und Wege finden, eine fraktale Grafik zweifarbig darzustellen, denn nur so können wir die hochauflösende Grafik des Spectrum ausnutzen. Dazu später mehr.

Das andere Problem ist das, wie wir unser Apfelmännchen im RAM aufbewahren. Das naheliegende wäre nun, ein zweidimensionales Array, also ein Zahlenfeld, zu definieren und die Werte darin zu speichern. Das ist im Prinzip auch richtig, hat aber einen Haken, der damit zusammenhängt, wie der Spectrum intern Zahlenwerte speichert. Egal, wie groß der Betrag einer Zahl im Endeffekt ist, sie nimmt immer 5 Bytes im RAM ein. Ein Zahlenfeld von 100*100 Punkten nimmt also ungefähr 50000 Bytes ein. (Es kommen noch einige Bytes für die Verwaltung des Feldes hinzu.) 50000 Bytes sind aber viel mehr als der 48K Speicher des Spectrum verkraften kann. (Das gilt auch für den 128k Spectrum.) Dabei ist ein Quadrat von 100*100 Punkten ein eher kleines Kästchen auf dem Bildschirm....

Ein Array kommt also nicht in Frage, wenn wir ein ansehnliches Bild darstellen wollen. Wir könnten natürlich auch darauf verzichten, die Werte abzuspeichern, und einfach nur für jeden Punkt entscheiden, ob er auf dem Bildschirm gesetzt wird oder nicht. Dann verbrauchen wir zwar keinen Speicher, und haben auch eine schöne Grafik, aber wir reduzieren die weite Spanne der errechneten Iterationen auf simple binäre Werte. Das ist zwar für die beiden Darstellungsmode, die ich in dieser Folge vorstellen will, ohne Bedeutung, aber andrerseits soll ja noch ein vierter Teil in dieser Reihe kommen, in dem eine weitere Methode der Apfelmännchendarstellung vorgestellt wird. Und warum sollten wir dafür dann alles noch einmal ausrechnen? (Immerhin kann der Spectrum bis zu eineinhalb Tagen an einer Grafik rechnen...ächz)

Glücklicherweise ist das aber alles gar kein ernstes Problem. Wir wissen, daß die Werte, die unsere Iterationsroutine ausrechnet, ganzzahlig sind, und bestenfalls einen Betrag von 20-25 annehmen können. So eine Zahl kann man mit einem Byte darstellen, und deshalb werden wir keine Variablen oder Felder deklarieren, sondern die Werte mit POKE direkt in den Speicher schreiben. Bevor wir das aber tun, müssen wir aber den Speicher vor unserem Basicprogramm schützen. Der Spectrum ist während des Programmablaufs nämlich ständig damit beschäftigt, hinter unserem Rücken Speicher für verschiedenste Arbeitsaufgaben zu beanspruchen und wieder freizugeben. Wenn wir unseren Speicherbereich nicht schützen würden, ist es nicht unwahrscheinlich, daß der Computer irgendwann in unserem Bild herumschreibt und die mühsam errechneten Werte unwissentlich durcheinanderbringt. Der Befehl CLEAR setzt dem Spectrum genau diese Grenze, und wir können beruhigt davon ausgehen, daß unser Bild sicher und warm liegt:

```
90 CLEAR 28000
```

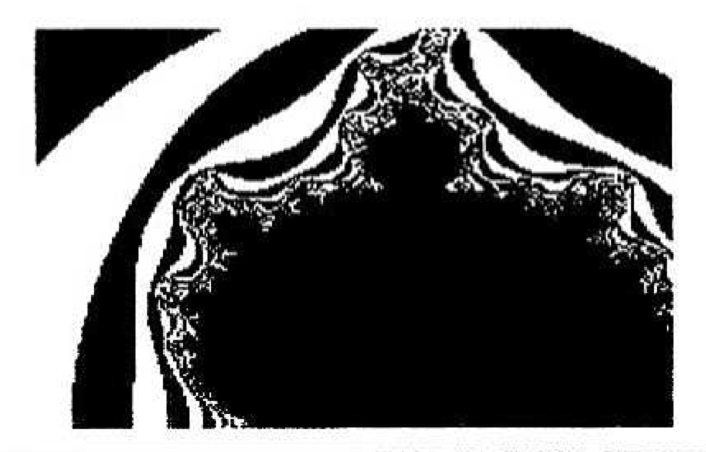
Den Wert 28000 habe ich mehr oder weniger willkürlich ausgesucht. Dadurch bleiben ungefähr 5K Speicher für Programm und Variablen übrig, was bei dem kurzen Mandelbrotprogramm völlig ausreicht, und wir haben rund 37000 Bytes für unser Bild. Das sind z.B. 210*175 Punkte, zwar nicht ganz ein Bildschirm voll, aber doch zumindest schon mal eine ganze Menge.

Jetzt geht's weiter. Bevor wir mit dem eigentlichen Programm beginnen, legen wir mal

eine ganze Menge Variablen fest, und kennzeichnen auch mit einer REM-Zeile, also mit einem Kommentar, was wir jetzt machen.

```
100 REM
110 REM * Parameter setzen *
120 REM
130 LET mx=200: LET my=150
140 LET pointer=28001
150 LET memstart=pointer
160 LET maxiter=10
170 LET sx=-2
180 LET sy=-1.5
190 LET ex=0.5
200 LET s=(ex-sx)/mx
210 LET n$="apfelbild"
```

Uff. Das war jetzt ja 'ne ganze Menge, und vielleicht ist auch nicht jede Variable klar verständlich. Aber mal von vorne: mx und my stehen für "maximales x" und "maximales y" und sagen aus, wie groß unser Bild in x- und y-Richtung wird, nämlich genau 200*150 Punkte. Pointer und memstart sind Variablen die für die POKEs in den Speicher gebraucht werden, und die ich weiter unten noch ausführlicher erkläre. Maxiter haben wir schon in der letzten Folge kennengelernt, diese Variable sagt dem Spectrum, wann er mit dem Rechnen aufhören kann: in diesem Fall geben wir nach 10 Durchläufen der Iterationsschleife auf. Sx und sy sind die Startkoordinaten in der Mandelbrotmenge. Ex ist die letzte x-Koordinate in der Mandelbrotmenge. Jetzt wird es ein bißchen kompliziert: wir haben ja eine gegebene Bildgröße, in diesem Fall 200*150, und wir müssen den Ausschnitt aus der Mandelbrotgrafik mit dieser Bildgröße in Einklang bringen. Wir teilen also zuerst die Strecke zischen sx und ex in soviele Punkte ein, wie unser Bild breit ist, in diesem Fall also in 200 Punkte. Damit haben wir die Schrittweite, mit der der Spectrum bei der Berechnung unseres Bildes durch die Apfelmenge schreiten wird. Aufmerksame Geister werden bemerkt haben, daß die ey Koordinate oben nicht definiert worden ist. Das ist auch gar nicht nötig, denn diese Koordinate ergibt sich ausgehend von der sy-Koordinate aus der Bildgröße in der y-Achse mal der Schrittweite. Dieses Verfahren hat den Vorteil, daß unser fertiges Bild nicht gestaucht oder gedehnt wird, sondern daß die Verhältnisse immer stimmen. Die Variable n$ zeigt schließlich an, unter welchem Namen die Daten auf dem Datenträger (Diskette, Microdrive oder Kassette ist egal) gespeichert werden.



```
220 POKE pointer, mx: poke
    pointer+1, my
230 POKE pointer+2,maxiter
240 LET pointer=pointer +3
```

Das sind jetzt die ersten Daten unseres Bildes. Pointer hatte ja auf das erste Byte hinter der Speichergrenze für den Spectrum gezeigt. Die ersten drei Bytes füllen wir mit der x- und der y-Größe. Das ist unbedingt notwendig, da wir sonst hinterher nicht mehr wissen, wie die Seitenverhältnisse in unserem Bild waren. 30000 Punkte könnten ja beispielsweise ein Bild von 150*200 Punkten sein, aber auch 300*100 Punkte oder 240*125 Punkte. Den Wert maxiter brauchen wir vielleicht noch für ausgefeiltere Darstellungstechniken. Es ist sicherer, den Wert zur Hand zu haben. In Zeile 240 wird der Speicherzeiger jetzt auf die erste Stelle nach diesen Informationen gerückt.

```
250 REM
260 REM * Hauptschleife *
270 REM
280 FOR a=1 to mx
290 FOR b=1 to my
300 GOSUB 450
310 POKE pointer,iter
320 LET pointer=pointer+1
330 IF iter/2 = INT(iter/2) THEN
    PLOT a,b
340 LET sy=sy+s
350 NEXT b
360 LET sx=sx+s
370 LET sy=sy-my*s
380 NEXT a
```

So. Das ist das ganze Hauptprogramm zur Berechnung und Darstellung des Apfelmännchens. Wie, das ist alles? Ja, das ist alles. Aber eins nach dem anderen... In Zeile 280 und 290 werden zwei FOR-NEXT Schleifen definiert. Variable a in Zeile 280 zählt die x-Koordinaten des Bildes. Für jeden einzelnen Punkt in x-Richtung werden dann in Zeile 290 folgende alle Koordinaten in y-Richtung durchlaufen, so daß insgesamt ein Rechteck von den Ausmaßen mx*my abgeschritten wird. Direkt in Zeile 300 folgt ein Aufruf der Iterationsroutine, die in der letzten Folge besprochen worden ist. Dieser Routine wurden ja die Werte sx und sy übergeben und sie gibt den Wert iter zurück. Das Ergebnis wird dann, wie oben besprochen, in die Speicheradresse pointer geladen, und pointer um eins erhöht.

Jetzt kommt in Zeile 330 endlich die erste Methode, die Grafik mit nur zwei Farben darzustellen. (Mein Gott, so ein Gedöhns und dann nur eine lausige Basiczeile.) Bei dieser Methode wird der Wert iter durch zwei geteilt und mit dem gerundeten Wert von iter/2 verglichen. Das heißt nichts anderes als zu überprüfen, ob iter eine gerade oder ungerade Zahl ist. Bei geraden Zahlen wird der entsprechende Punkt eingefärbt, ansonsten weiß gelassen. Auf diese Weise kann man auf der fertigen Grafik sehr schön die Verläufe der Iterationsschichten erkennen. Unsichtbar bleiben allerdings die Beträge, es ist nicht einmal zu sehen, ob ein Bildpunkt 2 mal oder 1000 mal iteriert ist. Um diesem Manko auf eine etwas plumpe Art und Weise abzuhelfen können wir Zeile 330 verändern:

```
330 IF iter > maxiter/2 THEN
    PLOT a,b
```

Jetzt werden nur die Punkte eingefärbt, die öfter als die Hälfte der maximal zulässigen Iterationsschritte durchgelaufen sind. Natürlich läßt sich dieses Verhältnis beliebig ändern. Insgesamt sind auf diese Art und Weise dargestellte Bilder weniger attraktiv als die mit der obigen Methode gezeichneten. Sie haben aber den Vorteil, daß man die Mandelbrotmenge selber etwas genauer Darstellen kann. Wenn bei der ersten Methode, die fertigen Bilder einer Meersebrandung gleichen, meint man bei der zweiten, urtümliche Kraken oder Amöben vor sich zu sehen...

Sei's drum. Jetzt haben wir wichtigeres zu tun, wir müssen nämlich die beiden FOR-NEXT Schleifen schließen. Das können wir aber nicht so ohne weiteres tun, denn wir müssen noch die Ausgangswerte für die Iterationsroutine, sx und sy, verändern. Innerhalb der b-Schleife in y-Richtung wird dies einfach dadurch getan, daß der Schrittwert s zu sy addiert wird. Dann kann die Schleife in Zeile 350 mit einem NEXT geschlossen werden. In Zeile 360 wird auf gleiche Weise mit der sx-Koordinate umgegangen. Nur darf jetzt noch nicht die FOR-NEXT Schleife geschlossen werden, sondern der sy-Wert muß zunächst zurückgesetzt werden. Wenn wir dies unterlassen (und das ist mir bei der ersten Version des Programms natürlich passiert), berechnen wir nämlich kein Rechteck, sondern eine Gebilde, daß man vielleicht am besten mit einer Art Treppe vergleichen kann. Jedenfalls würden die zu berechnenden Koordinaten mit hoher Geschwindigkeit ins Ungewisse sausen und hätten mit dem Bild, das wir eigentlich haben wollten, nur wenig zu tun. Zeile 370 erledigt das Problem, und in Zeile 380 wird die Schleife geschlossen.

Jetzt haben wir das Bild im Speicher und auf dem Bildschirm. Alles, was noch zu tun ist, wäre, die errechneten Werte auch auf dem Massenspeicher unserer Wahl abzuspeichern. Dazu fügen wir folgende Programmzeilen ein.

```
390 PAUSE 0
400 SAVE d1;n$ CODE memstart,
    pointer
410 STOP
```

Zeile 390 ist nicht unbedingt nötig. Es ist aber von Vorteil mit PAUSE 0 auf den Tastendruck des Computerbenutzers zu warten, denn es kann ja sein, daß der Spectrum mitten in der Nacht mit dem Rechnen fertig ist, und wer soll dann die richtige Diskette einlegen oder das Tonband starten? Zeile 400 ist hier in der +D Syntax angegeben. Wenn man den Code-Block auf Kassette oder auf dem Microdrive abspeichern will, ist lediglich die im Handbuch angegebene Syntax zu verwenden. Zeile 410 dient schließlich dazu, das Programm anzuhalten, bevor es unerlaubterweise in das Iterationsunterprogramm hineingerät.

Damit wären wir am Ende des dritten Teils angelangt. Im vierten und letzten Teil der Reihe werde ich eine Methode erläutern, die Mandelbrotgrafik perspektivisch darzustellen, damit die unterschiedlichen Iterationswerte auch in der Darstellung erhalten bleiben.

Es ist natürlich klar, daß die Basic-Sprache für ein solches Unterfangen wie die Berechnung einer fraktalen Grafik höchst unzureichend sein muß. Das Basic des Spectrum ist für einen 8-Bit Rechner zwar recht schnell, aber die Rechenoperationen dauern auch bei einer bescheidenen Iterationstiefe noch viel zu lange. Ich habe aber trotzdem auf beschleunigende Maschinenroutinen verzichet, weil ich demonstrieren wollte, daß selbst mit einfachsten Mitteln interessante Resultate auf dem Spektrum zu erzielen sind. Es gibt nicht den geringsten Grund auf das reine Basic herunterzusehen, so wie es manche "Hacker" tun. Das eigentlich reizvolle am Programmieren ist, sich Schritt für Schritt zu überlegen, wie ein Problem in Einzelaufgaben aufzuteilen und mit einfachsten Mitteln zu lösen ist. Das, was dabei herauskommt ist eine Art Kunstwerk, ein schöpferischer Akt, der weit mehr Freude und Befriedigung bringen kann, als das bloße Starten eines vorgefertigten Programms.

Hier noch einmal das ganze Programm zusammengefaßt:

```
10 REM
20 REM * Fraktale 2.0 *
30 REM
40 REM 1997 von Nele
50 REM
60 REM CLEAR auf einen sicheren Wert
   setzen:
```




```
 70 REM
 90 CLEAR 28000
100 REM
110 REM * Parameter setzen *
120 REM
130 LET mx=200: LET my=150
140 LET pointer=28001
150 LET memstart=pointer
160 LET maxiter=10
170 LET sx=-2
180 LET sy=-1.5
190 LET ex=0.5
200 LET s=(ex-sx)/mx
210 LET n$="apfelbild"
220 POKE pointer, mx: POKE pointer
    +1, my
230 POKE pointer+2,maxiter
240 LET pointer=pointer +3
250 REM
260 REM * Hauptschleife *
270 REM
280 FOR a=1 to mx
290 FOR b=1 to my
300 GOSUB 450
310 POKE pointer,iter
320 LET pointer=pointer+1
330 IF iter/2 = INT(iter/2) THEN
    PLOT a,b
340 LET sy=sy+s
350 NEXT b
360 LET sx=sx+s
370 LET sy=sy-my*s
380 NEXT a
390 PAUSE 0
400 SAVE d1;n$ CODE memstart,
    pointer
410 STOP
420 REM
430 REM * Iterationen *
440 REM
450 LET x=0: LET y=0
460 LET iter=0
470 FOR i=1 TO maxiter
480 LET xn=xx*x-y*y+sx
490 LET y=2*x*y+sy
500 LET x=xn
510 IF SQR (x*x+y*y)>2 THEN LET
    iter=i: LET i=maxier
520 NEXT i: RETURN
```

Nele Abels-Ludwig, Am Mühlgraben 4
35037 Marburg, Tel. 06421/210272

Deutsche Übersetzung des TECHINFO.DOC des Sinclair ZX Spectrum Emulator 'Z80' v3.05 - 11/11/96 - von G.A. Lunter

Vorwort zu dieser Übersetzung: Ich habe - soweit mein Englisch reicht - diesen, meiner Meinung nach sehr interessanten Text ins Deutsche übersetzt. Für Fehler, Fehl-Interpretationen oder ähnliches kann ich keine Gewähr übernehmen, würde mich aber freuen, wenn mir in einem solchen Fall jemand Bescheid geben könnte! Danke, Bernhard.

5. TECHNISCHE INFORMATIONEN (5)

5.7 Das Multiface 128 (Fortsetzung)

Beachte das ein NMI auch das Disciple ROM einblendet, doch da das Multiface Vorrang hat, werden die Multiface Routinen abgearbeitet.
Genau genommen sieht es so aus, als ob das Disciple ROM überhaupt nicht eingeblendet wird (doch möglicherweise blenden es auch die Multiface Routinen selbst aus).
Der Multiface Speicher wird über ein Lesen von Port #3F ausgeblendet.
Das Multiface legt keine Bits auf den Bus.

5.8 Das AMX Maus Interface

Das einzige Programm das ich kenne und das die AMX Maus unterstützt ist Art Studio, doch es ist so gut, das es für mich genug Grund ist, das AMX Maus Interface einzubauen.
Das AMX Interface benutzt ein Z80-PIO (programmierbares In/Out Interface).
Dieser Chip ist schon etwas kompliziert, und kann in mehreren Modi betrieben werden. Das AMX Interface benutzt nur Modus 1 (nebenbei bemerkt: Hier keine Verbindung mit IM 1), sodaß dies der einzige Modus ist, der emuliert wird. Die folgende Erläuterung bezieht sich nur auf Modus 1 und nur auf die I/O Adressen, die vom AMX Maus Interface benutzt werden.
Das PIO enthält zwei 8-Bit Interrupt Vector Register (VECA und VECB), zwei 2-Bit Modus Register (MODEA und MODEB), zwei 8-Bit Daten Register (DATAA und DATAB), und zwei 1-Bit Interrupt-Enable Latches (IEA und IEB). Die Daten Register können jederzeit über die IN Ports #1F und #3F abgetragt werden. (Beachte das #1F auch durch das Kempston Joystick Interface genutzt wird, und auch durch das Disciple Interface, und das #3F auch vom Multiface benutzt wird). Die anderen Register können nicht gelesen werden, sondern nur über die Nur-Schreibe-Register CTRLA und CTRLB gesetzt werden, auf die über OUT Port #5F und #7F zugegriffen wird:

Interrupt vector setzen:

Bit	7	6	5	4	3	2	1	0
WRITE	interrupt vector							0

IE Latch setzen:

Bit	7	6	5	4	3	2	1	0
WRITE	IE	x	x	x	0	x	x	1

PIO Modus setzen:

Bit	7	6	5	4	3	2	1	0
WRITE	MODE		x	x	1	1	1	1

Wenn IEA oder IEB gesetzt sind (1), werden die korrespondierenden (=zugehörigen) Interrupt-Sequenzen Interrupts erzeugen, wenn sie dazu durch die Hardware veranlaßt werden, und den korrespondierenden Interrupt Vector (=Zeiger) zur Interrupt-Zeit auf den Daten-Bus legen. Im Z80 Interrupt Modus 2 wird dies dann als Zeiger benutzt. Beachte, das Bit 0 des Zeigers immer 0 ist. Z80 emuliert nur das PIO Modus 1 Verhalten (Bit 7=0, Bit 6=1 wenn PIO Mode der gesetzt wird), und wird überhaupt nichts tun, wenn der Modus auf ein anderen Wert gesetzt wird.
Das AMX Interface generiert einen A Interrupt für jede kleine Bewegung (="Mickey"), die die Maus in X-Richtung erzeugt, und einen B Interrupt für jeden Mickey in Y-Richtung. Das Vorzeichen ("SIGN") jeder Richtung wird als 0 (positiv) oder als 1 (negativ) in den zugehörigen Daten-Registern gespeichert.
Ein zusätzlicher Port, IN Port #DF, liefert den Status der Maus-Taste. Er liefert #FF oder #00 zurck.
Die AMX wird wie folgt emuliert. Bei jedem 50 Hz Zyklus wird der Maus-Status überprüft, und ein PIO Interrupt emuliert, wenn nötig. Die IRET Anweisung wird ge-trapped, und wenn sie gefangen (caught) wird, werden wenn nötig mehr PIO Interrupts emuliert, nachdem der aktuelle IRET abgearbeitet wurde, mit einem Maximum von 32 Interrupts pro Zyklus pro Koordinate. Es wird darauf geachtet, daß der 50 Hz Zyklus Interrupt auch ausgeführt wird. Dieser letztgenannte Interrupt kann vom PIO Interrupt dadurch unterschieden werden, daß er #FF auf den Bus legt. Es ist außerdem wichtig, darauf zu achten, ob die erste Anweisung nach dem RETI ein HALT ist, diese wird dann ausgelassen.
Es sieht so als ob der PIO den Z80 für ein paar Takte, nachdem ein RETI erkannt wurde in Frieden laßt, bevor er einen weiteren Interrupt erzeugt.
Ein Reset-Signal setzt die IE Latches des PIO zurück.

5.9 Der Z80 Microprozessor

Der Z80 Prozessor ist wirklich gerade, und enthält meiner Kenntnis nach keine interessanten Fehler oder Ungereimtheiten. Trotzdem enthält er undokumentierte Fähigkeiten. Einige von diesen sind wirklich nützlich, und andere nicht, doch da viele Programme die nützlichen verwenden, und ein paar andere die häßlichen, habe ich versucht sie herauszufinden und so gut wie mir möglich zu emulieren. Es gibt einen anderen Z80 Emulator, gedacht als CP/M Emulator, welcher das Programm stoppt wenn ein undokumentierter Opcode auftritt. Ich denke nicht, daß dies einen Sinn macht.
ZILOG diktiert nicht das Gesetz, sondern die Programme, die den Prozessor benutzen tun das!
Im Abschnitt 5.1 stehen einige Informationen zu den Z80 Interrupt Timings in verschiedenen Modi.

Die meisten der Z80 Opcodes sind ein Byte lang, nicht mitgezählt wird dabei ein möglicher Byte oder Wort Operant. Die vier Opcodes CB, DD, ED und FD sind 'shifted' Opcodes: Sie ändern die Bedeutung der Opcodes, die nach ihnen folgen.
Es gibt 248 verschiedene CB Opcodes. Der Block CB 30 bis CB 37 ist normalerweise nicht auf der offiziellen Liste vorhanden. Diese Anweisungen, normalerweise mit dem Mnemonic SLL (Shift Left Logical) bezeichnet, schieben den Operanten nach links und Bit 0 immer zu 0. Bounder und Enduro Racer benutzen diese, um nur zwei zu nennen. Der SamRam Monitor kann diese disassemblieren und benutzt den Mnemonic SLL. Diese Anweisungen werden wirklich öfters benutzt.
Die DD und FD Opcodes stehen Anweisungen voran, die die IX und IY Register benutzen. Wenn man sich die Anweisungen genau ansieht sieht man wie sie funktionieren:

```
2A nn       LD HL,(nn)
DD 2A nn    LD IX,(nn)
7E          LD A,(HL)
DD 7E d     LD A,(IX+d)
```

Ein DD Opcode ändert einfach die Bedeutung des HL in der nächsten Anweisung.
Wenn ein Speicherbyte indirekt über HL adressiert wird, wie im zweiten Beispiel, wird ein Ersatz- ("Displacement-") Byte hinzugefügt. Weiterhin funktioniert die Anweisung einfach mit IX anstatt mit HL. (Eine notationelle Peinlichkeit in der Schreibweise, die nur Assembler und Disassembler-Autoren berührt: JP (HL) ist nicht indirekt; es sollte eigentlich JP HL heißen). Wenn ein DD Opcode vor einer Anweisung steht, die das HL Registerpaar überhaupt nicht benutzt, wird diese wie normal ausgeführt. Doch, wenn die Anweisung das H oder L Register benutzt, wird in diesem Fall die höherwertige oder niederwertigere Hälfte des IX Registers benutzt! Beispiel:

```
44          LD B,H
FD 44       LD B,IYh
```

Diese Typen von inoffiziellen Anweisungen werden in vielen Programmen benutzt. Nebenbei, viele DD oder FD Opcodes nacheinander sind effektiv wie NOPs, sie machen nichts außer wiederholt das Flag 'Ersetze HL durch IY' (oder IY) zu setzen und 4 T States lange zu dauern. Beachte, das der FD oder DD 'Opcode' als Teil einer Anweisung verstanden wird, sodaß der Z80 während der Ausführung eines solchen Blocks nicht unterbrochen werden kann. (Diese letzte Bemerkung wurde nicht überprüft, sie wurde von jemand von comp.sys.sinclair gemacht).
Wenn (Teile von) HL sowohl als Quelle aus auch als Ziel genutzt werden, sind die oben beschriebenen Regeln mehrdeutig. Diese speziellen Fälle sind:

LD H,(IX+d) und LD L,(IX+d), und LD IXl,IXl / LD IXl,IXh / LD IXh,IXl / LD IXh,IXh. Dinge wie LD IXh,(IX+d) oder LD H,IXh existieren nicht.

Zwei Anweisungen folgen nicht der DD/FD Regel. Diese sind EX DE,HL und EXX. Diese Anweisungen dauern 4 T States zur Abarbeitung, und werden damit meistens damit genutzt indem man ein Flag ändert, welches die Referenz zu den Registern wechselt; keine Daten werden wirklich bewegt, was erklärt, weshalb sie nicht mit IX oder IY zusammenarbeiten.
Auch die doppelt geshifteten Opcodes die mit DD CB und DD ED beginnen, folgen nicht der Standard-Regel. Wenn ein DD oder FD vor einer ED Anweisung steht, wird diese ignoriert. ED Anweisungen funktionieren nie mit den IX oder IY Registern. Mit den CB Anweisungen ist die Situation interessanter. Jede DD CB Anweisung arbeitet mit (IX+nn), kopiert aber auch das Resultat in das gewählte Register, außer wenn es (HL) ist.
Zum Beispiel:

```
CB CE          SET 0,(HL)
CB C0          SET 0,B
DD CB nn CE    SET 0,(IX+nn)
DD CB nn C0    SET 0,(IX+nn);
```

kopiert Resultat nach B.
(Die Informationen über die inoffiziellen CB Anweisungen bekam ich von Arnt Gulbrandsen, publiziert durch David Librik).
Es gibt eine Anzahl von inoffiziellen ED Anweisungen, doch keine davon sind sehr nutzvoll. Die ED Opcodes im Bereich von 00-3F und 80-FF (natürlich mit Ausnahme der Block Anweisungen) machen überhaupt nichts als 8 T States zu dauern und das R Register um 2 zu erhöhen. Die meisten der nicht aufgelisteten Opcodes im Bereich von 40-7F haben jedoch einen Effekt. Die komplette Liste: (* - nicht offiziell):

Opcode		Anweisung	Opcode		Anweisung
ED40		IN B,(C)	ED60		IN H,(C)
ED41		OUT (C),B	ED61		OUT (C),H
ED42		SBC HL,BC	ED62		SBC HL,HL
ED43		LD (nn),BC	ED63	#	LD (nn),HL
ED44		NEG	ED64	*	NEG
ED45		RETN	ED65	*	RET
ED46		IM 0	ED66	*	IM 0
ED47		LD I,A	ED67		RRD
ED48		IN C,(C)	ED68		IN L,(C)
ED49		OUT (C),C	ED69		OUT (C),L
ED4A		ADC HL,BC	ED6A		ADC HL,HL
ED4B		LD BC,(nn)	ED6B	#	LD HL,(nn)
ED4C	*	NEG	ED6C	*	NEG
ED4D		RETI	ED6D	*	RET
ED4E	*	IM 0	ED6E	*	IM 0
ED4F		LD R,A	ED6F		RLD
ED50		IN D,(C)	ED70	#	IN (C)
ED51		OUT (C),D	ED71	*	OUT (C),0
ED52		SBC HL,DE	ED72		SBC HL,SP
ED53		LD (nn),DE	ED73		LD (nn),SP
ED54	*	NEG	ED74	*	NEG
ED55	*	RET	ED75	*	RET
ED56		IM 1	ED76	*	IM 1
ED57		LD A,I	ED77	*	NOP
ED58		IN E,(C)	ED78		IN A,(C)
ED59		OUT (C),E	ED79		OUT (C),A
ED5A		ADC HL,DE	ED7A		ADC HL,SP
ED5B		LD DE,(nn)	ED7B		LD SP,(nn)
ED5C	*	NEG	ED7C	*	NEG
ED5D	*	RET	ED7D	*	RET
ED5E		IM 2	ED7E	*	IM 2
ED5F		LD A,R	ED7F	*	NOP

Guenter Woigk bemerkte dazu, das die drei Anweisungen, die mit # markiert sind (welche zuvor mit * markiert waren) von ZiLog dokumentiert, aber normalerweise nicht gebräuchlich sind.
Die ED63 und ED6B Anweisungen haben kürzere und schnellere Entsprechungen.
Die ED70 Anweisung liest von Port (C), genau wie die anderen Anweisungen, wirft aber das Resultat weg. Es verändert nicht die Flags in der gleichen Manier wie die anderen IN Anweisungen. Die ED71 Anweisung OUTet interessanterweise ein Byte 0 auf den Port (C). Diese Anweisungen 'sollten', nach der Regel des Sets der Anweisungen, (HL) als Operanten nutzen, doch aus Sicht des Prozessors ist der Speicherzugriff oder ein Zugriff auf I/O Geräte dasselbe, mit Ausnahme der Aktivierung der /IORQ Leitung anstatt der /MREQ line, und da der Z80 innerhalb einer Anweisung nicht zweifach auf den Speicher zugreift (außer bei Mißachtung des Anweisungsfetch natürlich) kann es das Daten-Byte nicht bearbeiten und speichern. (Ein Tip in dieser Richtung ist das, auch wenn die NOP-Synonyme LD B,B, LD C,C etc. existieren, LD (HL),(HL) nicht vorhanden ist mit einer HALT Anweisung ersetzt wird).
Die Anweisungen ED 4E und ED 6E sind IM 0 Äquivalente: Wenn FF zur Interrupt-Zeit auf den

Bus gelegt wird (physikalisch), führt der Spectrum normal mit der Abarbeitung fort, wenngleich er abstürzt, falls eine EF (RST #28) Anweisung auf den Bus gelegt wird, wie er es auch tun wird, wenn der Z80 im offiziellen Interrupt-Modus 0 ist. Im IM 1 führt der Z80 einfach einen RST #38 (Opcode FF) aus, egal was auf dem Bus liegt.
Die RETI Anweisung funktioniert genau äquivalent zur RET Anweisung. Sie wird nur dazu benutzt, einem externen Hardware Gerät das Ende einer Interrupt Routine anzuzeigen (dies: der Z80 PIO). Andererseits ist RETN dadurch verschieden zu RET, das es IFF1 auf den aktuellen Wert von IFF2 setzt. Nun sind IFF1 und IFF2 normalerweise gleich (und werden gleich, nach DI und EI und nachdem ein maskierbarer Interrupt akzeptiert wurde). Sie sind nur verschieden, wenn bei ermöglichtem (ENABLED) Interrupts ein NMI auftritt: dann ist IFF1 ausgeschaltet, und IFF2, welches den vorherigen Status des Interrupt-Flip-Flops enthält, ist eingeschaltet, was signalisiert, das die Interrupts vor dem nicht-maskierbaren Interrupt eingeschaltet waren. Da der Status von IFF2 durch LD A,R und LD A,I gelesen werden kann, wird die RETN Anweisung nicht sehr oft in der Spectrum ROM Software genutzt, und ist eigentlich nutzlos in normaler Software. Mit anderen Worten, habe ich auch nicht versucht herauszufinden, ob die inoffiziellen RET's entweder RETI's oder RETN's sind. Man kann aber aus dem gelernten eigene Schlüsse ziehen.

Über das R Register. Dieses ist nicht wirklich eine undokummentierte Fähigkeit, wenn auch präzise Erklärungen dazu schwierig zu finden sind. Das R Register ist ein Zähler, der mit jeder Anweisung erhöht wird, wobei DD, FD, ED und CB als separate Anweisungen angesehen werden. So geshiftete Anweisungen erhöhen R um zwei. Es gibt eine Ausnahme:
Doppelt-geshiftete Opcodes, diejenigen mit DDCB und FDCB, erhöhen R auch mit um zwei. LDI erhöht R um zwei, LDIR erhöht es um 2 mal BC, wie auch LDDR etc. Die Sequenz LD R,A / LD A,R erhöht A um zwei.
Das höchste Bit des R Registers ändert sich niemals (außer natürlich durch LD R,A). Das liegt daran, das in den alten Tagen jeder 16 Kbit Chips verwendete. Innerhalb des Chips sind die Bits in einer 128x128 Matrix angeordnet, und benötigen einen 7 Bit Refresh Zyklus. Möglicherweise aus diesem Grund hat ZiLOG entschieden nur die unteren 7 Bits zu zählen.
Wenn die R Register-Emulation eingeschaltet ist, verhalt sich das R Register wie auf einem realen Spectrum; wenn sie ausgeschaltet ist wird es sich (außer das oberste Bit) wie ein Zufallsgenerator verhalten.
Man kann es einfach überprüfen, daß das R Register wirklich entscheidend für dem Speicher Refresh ist. Assembliere das Programm:

```
     ORG 32768
     DI
     LD B,0
L1   XOR A
     LD R,A
     DEC HL
     LD A,H
     OR L
     JR NZ,L1
     DJNZ L1
     EI
     RET
```

Es dauert etwa drei Minuten zum durchlaufen. Schaue nun in die oberen 32K des Speichers, zum Beispiel in die UDG-Graphics. Sie sind zerstört. Nur die ersten wenigen Bytes jedes 256 Byte Blocks wird weiterhin Nullen enthalten weil sie während der Ausführung der Schleife refreshed werden. Die ULA wacht über den Refresh der unteren 16K. (Dieses Beispiel wird allerdings nicht auf dem Emulator funktionieren !).
Dann ist da noch eine andere schwarze Ecke des Z80 welches seinen Effekt in Programmen wie Sabre Wulf, Ghosts'n Goblins und Speedlock zeigt. Das Geheimnis der undokummentierten Flags!
Bit 3 und 5 des F Registers werden nicht benutzt. Sie können Informationen beinhalten wie man auch durch die Benutzung von PUSH AF und POP AF prüfen kann. Weiterhin wechselt deren Wert manchmal. Ich fand die folgende empirische Regel: Der Wert der Bits 7, 5 und 3 folgt den Werten der korrespondierenden Bits des letzten 8 Bit Resultat einer Anweisung die die normalen Flags setzt.
Z.B., sind nach einem ADD A,B diese Bits identisch zu den Bits des A Registers. (Bit 7 von F ist das Vorzeichen ("Sign") Flag, und folgt der Regel exakt). Eine Ausnahme ist die CP x Anweisung (x=Register, (HL) oder ein direktes Argument). In diesem Fall werden die Bits vom Argument kopiert.
Wenn die Anweisung eine ist, welche als Operant ein 16 Bit Wort hat, sind die 8 Bits der Regel die obersten 8 Bits des 16 Bit Resultats - das war zu erwarten, da das S Flag von Bit 15 gewonnen wird.
Ghosts'n Goblins benutzt die undokumentierten Flags durch einen Programmierfehler. Das Nashorn läuft rückwärts oder bleibt in kleinen Kreisen laufend in einer Ecke, wenn (das in diesem Fall undokumentierte) Verhalten des Sign Flags in der BIT Anweisung nicht richtig ist. Ich zitiere (Ein faszinierendes Stück von Maschinen-Code!):

```
AD86   DD CB 06 7E    BIT 7,(IX+6)
AD8A   F2 8F AD       JP P,#AD8F
```

- Wird fortgesetzt -

Bernhard LUTZ, Hammerstr. 12, 76756 Bellheim
Tel. 07272/77372 (b. Sprenger, Mo-Do ab 18 Uhr)
Fax/AB/Mailbox: 07272/92108
e-mail: luzie@t-online.de



Nichts bestimt so sehr die Überlebensdauer eines Rechners, wie die für diesen angebotene Software. Nun hatten wir in der Vergangenheit in dieser Hinsicht für unseren Speccy ja keine Probleme. Nur mit neueren Programmen haperts, da es keine den Speccy unterstützenden Softwarehäuser mehr gibt. Na ja, fast keine, denn es gibt doch noch welche, aber die tragen nicht mehr die Namen Ultimate, Mastertronic, Ocean etc. Und sie kommen nicht mehr aus good ol' England.
So bekamen wir dieser Tage ein Spiel aus der (ehemaligen?) Slowakei zur Ansicht, das eine nähere Betrachtung verdient und einen einprägsamen Namen hat:

QUADRAX

Hersteller/Jahr:	**Ultrasoft 1994**
Vertrieb:	**SINTECH, Thomas Eberle** **Gastäckerstraße 23** **Tel./Fax: 0711/775033**
Preis:	**10 DM + Versandkosten**

Zur Vorgeschichte: Tief in der Wüste Sahara steht der sagenumwobene Tempel der Sonne. Es wird erzählt, daß keiner von denen die hineingingen, je wieder nach Hause fanden. Ältere Legenden sagen, daß auf der höchsten Ebene des Tempels die Göttin der Sonne lebt.
Zum Spielverlauf: Hier setzt nun die Aufgabe des Spielers (oder zweier Spieler!) ein, mit 2 Spielfiguren durch alle Räume des Tempels zu gehen und die höchste Ebene zu erreichen. Klingt einfach... na, wartet es ab.
Jeder Raum hat nur einen Ausgang, und den muß man mit beiden Spielfiguren erreichen. Dabei gibt es manch knifflige Puzzel zu lösen, Schalter zu bedienen und Fahrstühle zu benutzen... kurzum, das Spiel wird nicht langweilig und am besten notiert man sich die erarbeiteten Paßwörter für weitere Spiele.
Leider bin ich bisher noch nicht allzuweit gekommen. Dennoch möchte ich sagen, daß das Spiel gut umgesetzt und die Animation unserer Spielfiguren (Helden) gelungen ist. Dem Spiel liegt eine deutsche Anleitung bei, die es einfach macht, bei der Steuerung auf Angaben wie Hore (hoch) oder Prepinac (umschalten) die richtigen Tasten einzugeben.
QUADRAX läuft auf allen Spectrum-Modellen, außer auf dem nicht umgebauten +2A. Sollte jemand zufällig einen Delta oder Didaktik (Spectrum Clones) besitzen - auch kein Problem. Ein Problem gibt es dafür aber völlig unerwartet, wenn man das Programm laden will, jedoch ein Diskinterface (evtl. auch andere) angeschlossen hat. Ein spezielles Kopierschutzprogramm verhindert dann bereits nach dem Vorlader jedes weitere Laden. Dies ist auch der Grund dafür, warum ich euch keinen Screenshot hier zeigen kann.
Sieht man jedoch davon einmal ab, so erhält man für wenig Geld ein Spiel, an dem man je nach Können Stunden, Tage oder Monate verbringen kann.

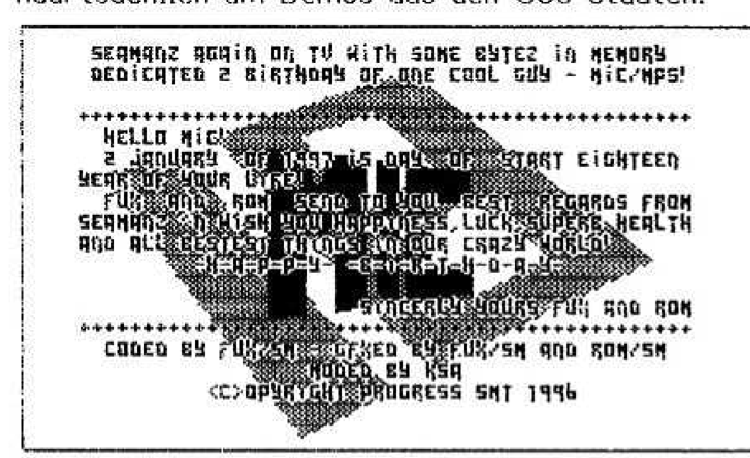
SPECCY CD 97

Vertrieb:	**SINTECH, Thomas Eberle** **Gastäckerstraße 23** **Tel./Fax: 0711/775033**
Preis:	**25 DM + Versandkosten**
System:	**PC (am besten 386 oder höher)**

Mein erster Gedanke war: Mensch, daß ich das noch erlebe! Aber es ist wahr: SINTECH unter der Leitung von Thomas Eberle hat nun auch den Vertrieb dieser CD für die Emulatorfreunde mit ins Programm aufgenommen.
Die CD enthält über 3000 Spectrum Programme, Anleitungen, Tips und Tricks und natürlich auch einen Emulator. Außerdem bekomt man noch 1300 C 64 Programme, ebenfalls inclusive Emulator sozusagen als Beigabe.

Demo-Szene

In letzter Zeit war es in der Demo-Szene recht still geworden. Bis ebenfalls dieser Tage Post aus England, genauer gesagt von Matthew Westcott, kam. Besten Dank dafür Matthew und - Dein deutsch ist wirklich erstklassig.
Wenngleich nur wenige Demos auf der beiliegenden Diskette waren, lohnt es sich doch, auf diese etwas näher einzugehen. Es handelt sich dabei hauptsächlich um Demos aus den GUS-Staaten.



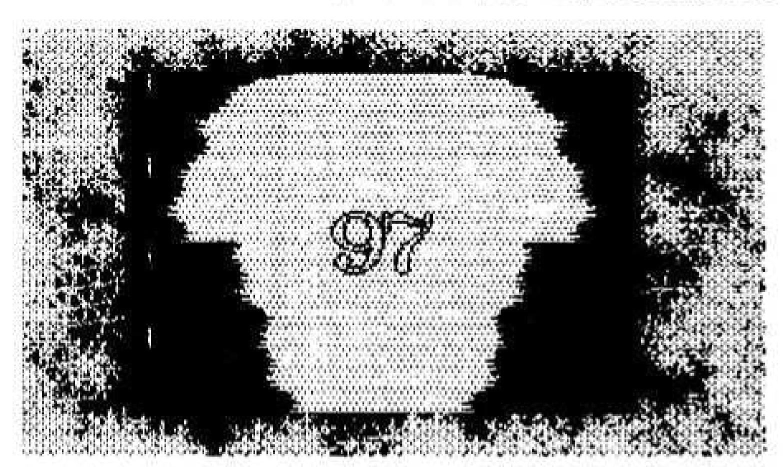
MIC (Progress)
Dies ist eines der typischen russischen "Minimaldemos", mit denen man dort gerne befreundete Coder zum Geburtstag grüßt. So auch in diesem Fall. Denn außer einem drehenden MIC-Logo gibt es nichts weiter zu sehen. Das wär's denn auch eigentlich schon, wäre da nicht noch diese mich immer wieder ansprechende russische Musik im Hintergrund. Und die ist, wie schon so oft, einfach excellent.

Happy Birthday Alex (Progress)
Zu diesem Demo gibt es eigentlich nichts anderes zu sagen, als zu "MIC", nur das es sich diesmal um ein sich drehendes ineinander verschlungenes Progress-Logo handelt vor dem ein Stern auf und ab hüpft und die Tatsache, daß dieses Demo auch noch den eigentümlichen ULA-Snow produziert, der sich beim Lesen des Scrolltextes manchmal sehr störend bemerkbar macht.

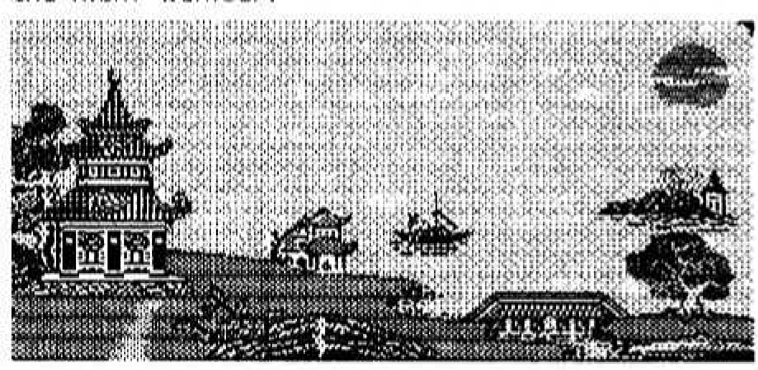


Brain Surgery (Rush)
Kommen wir zu einem der besseren Demos (ach was sag ich, das Demo ist super). Brain Surgery ist die "Konsequenz" zu Mental Masturbation. Und bietet wieder mal Effekte vom Feinsten.
Nach einer Einleitung wird man mit schnellen Farbverläufen (Zoids) überrascht. Der nachfolgende Teil ist der vielleicht impressivste. Farbige Sinuswellen rollen ohne Attribute-Clashes über den Bildschirm (2-Bitplanes Multisines), wahrlich etwas fürs Auge. Der folgende Realtime Plasmaeffekt ist zwar auch hübsch anzusehen, aber nicht neu. Lustig dagegen ist das gezoomte Portrait von 'Joker'. Und dem abschließenden Greetinx-Part ist der obige Screenshot entnommen. Wir freuen uns natürlich über die Grüße. Greetings back to you, guyz!

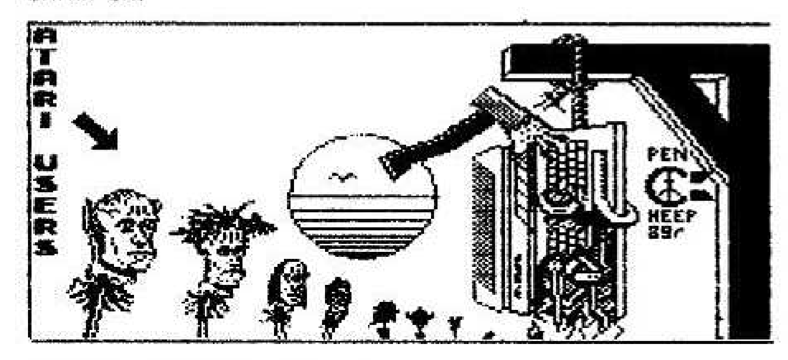
Yolka 97 (Rush)
Der Screenshot rechts oben ist ein echter Glücksfall, denn das Demo stürzt auf unserem +2 völlig unkontrollierbar ab, mal nach wenigen Sekunden, mal nach einer Minute. Wenn es also läuft, sieht man wieder einen netten Plasmaeffekt (mit ULA-Snow) und auch die Musik scheint irgendwie falsch zu laufen. Vielleicht ein Kopierfehler auf der Diskette?



Always On My Mind (Coolguys Limited)
Samplesound mit hübschem Screen - nicht mehr und nicht weniger.



Atari Shit 1 + 2 (The Cybernoids)
Diese beiden Demos sind schon älteren Datums und wurden von Matthew auf einer Spectrum-Emulator-CD entdeckt. Auch hierzu gibt es nicht viel zu sagen. Zu einem Screen kann man sich in beiden Demos mehrere Lieder aus 128K Spielen anhören.



PD-Szene

Steve Dotman Designer (Matthew Westcott)
Natürlich wollen wir nicht verschweigen, das Matthew auch eine verbesserte Version des an dieser Stelle schon einmal besprochenen "Steve Dotman Designers" mitgeschickt hat. Es gibt ab Jetzt nur noch eine Version für 48K und 128K, jedoch jeweils für Kassette bzw. Plus D. Matthew sucht immer noch nach Screes für sein geplantes Spiel! Na, hat keiner Lust?